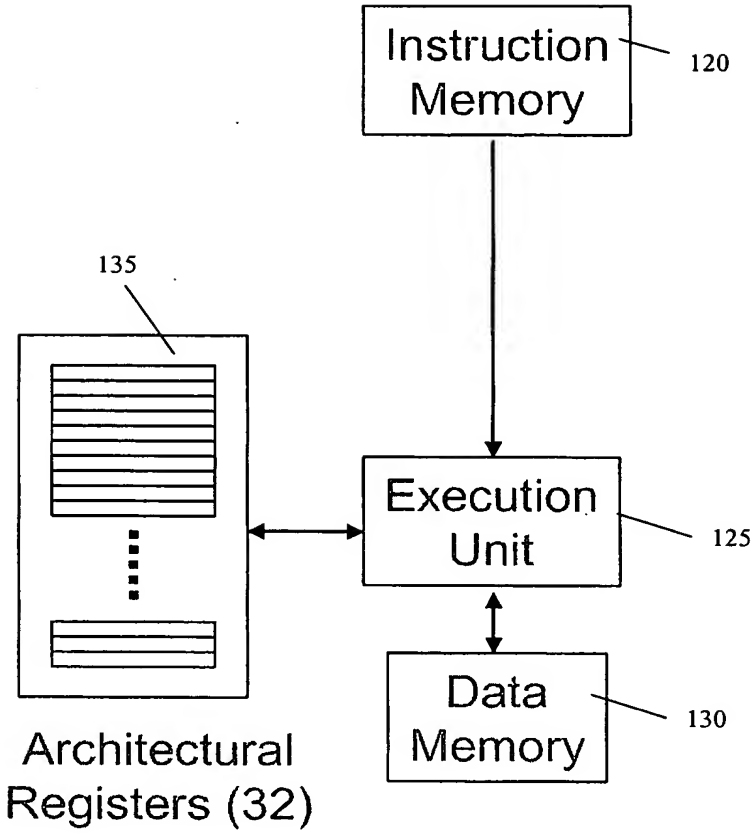
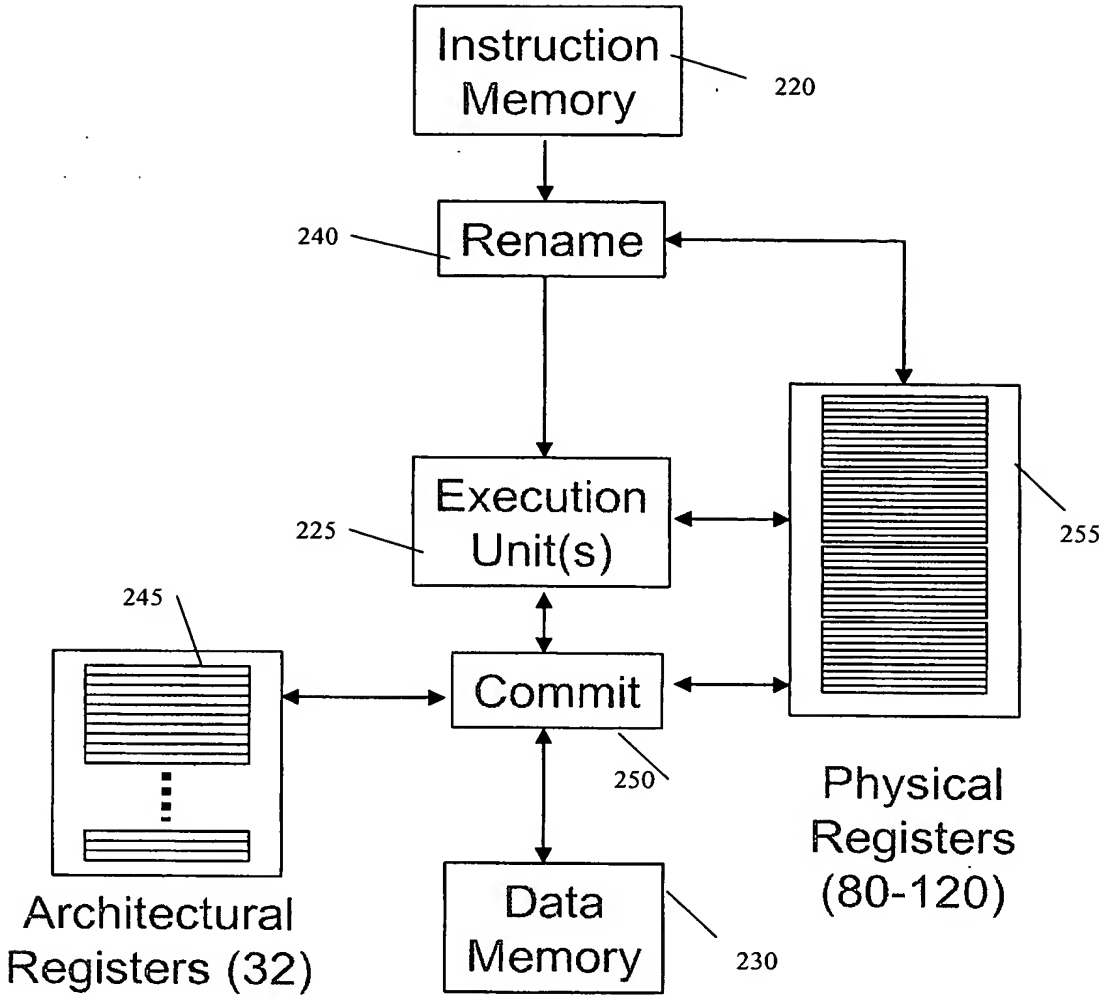
Fig. 1



Prior Art

Fig. 2



Prior Art

Fig. 3

```
.....
.....
Load R1 = A
Load R2 = B
; Use R1
.....; ran out of registers
Spill R1 back to location A
Load R1 = C
R2 = R2 + R1 ; B=B+C
.....
.....
Spill R1 to location C
Load R1 =  A
R2 = R2 * R1 ; B=B*A
.....
.....
```

Fig. 4

```
; nameLev R1, 0 is implicit
.....
.....
Load R1 = A ; first instance of R1
Load R2 = B
; use R1
.....
; ran out of registers
nameLev R1,1    ; R1 means "second instance" of R1
Load R1 = C ; second instance of R1
R2 = R2 + R1; second instance of R1
.....
.....
nameLev R1, 0    ; R1 means "first instance" of R1
R2 = R2 * R1; first instance of R1
.....
.....
```

Fig. 5

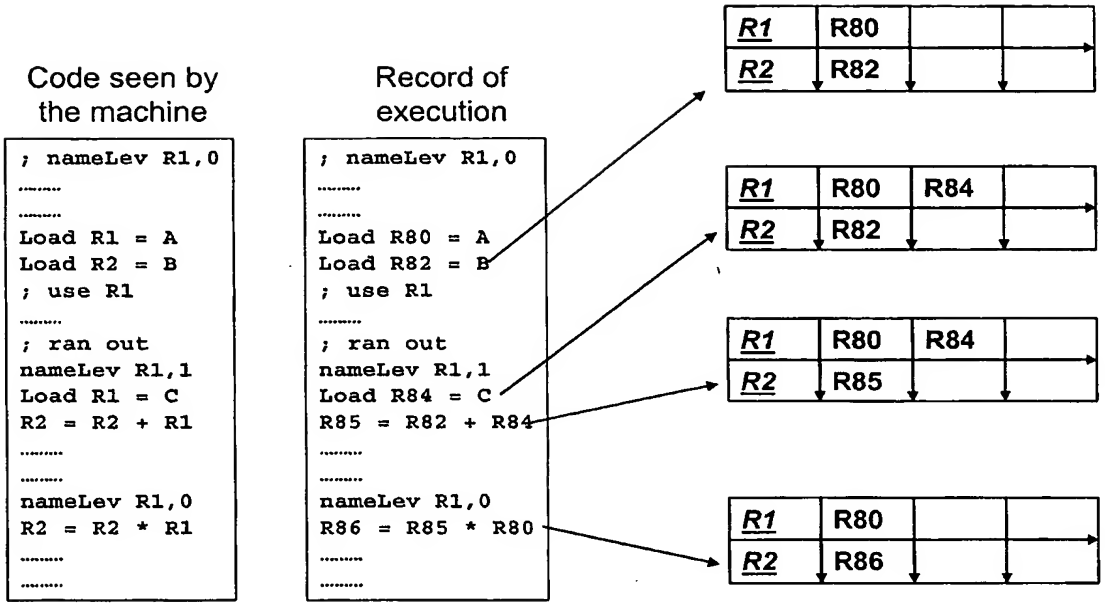

Code seen by the machine
; nameLev R1,0
.....
.....
Load R1 = A
Load R2 = B
; use R1
.....
; ran out
nameLev R1,1
Load R1 = C
R2 = R2 + R1
.....
.....
nameLev R1,0
R2 = R2 * R1
.....
.....
Record of execution
; nameLev R1,0
.....
.....
Load R80 = A
Load R82 = B
; use R1
.....
; ran out
nameLev R1,1
Load R84 = C
R85 = R82 + R84
.....
.....
nameLev R1,0
R86 = R85 * R80
.....
.....
R1
R2
R80
R82
R1
R2
R80
R84
R82
R1
R2
R80
R84
R85
R1
R2
R80
R86

Fig. 6

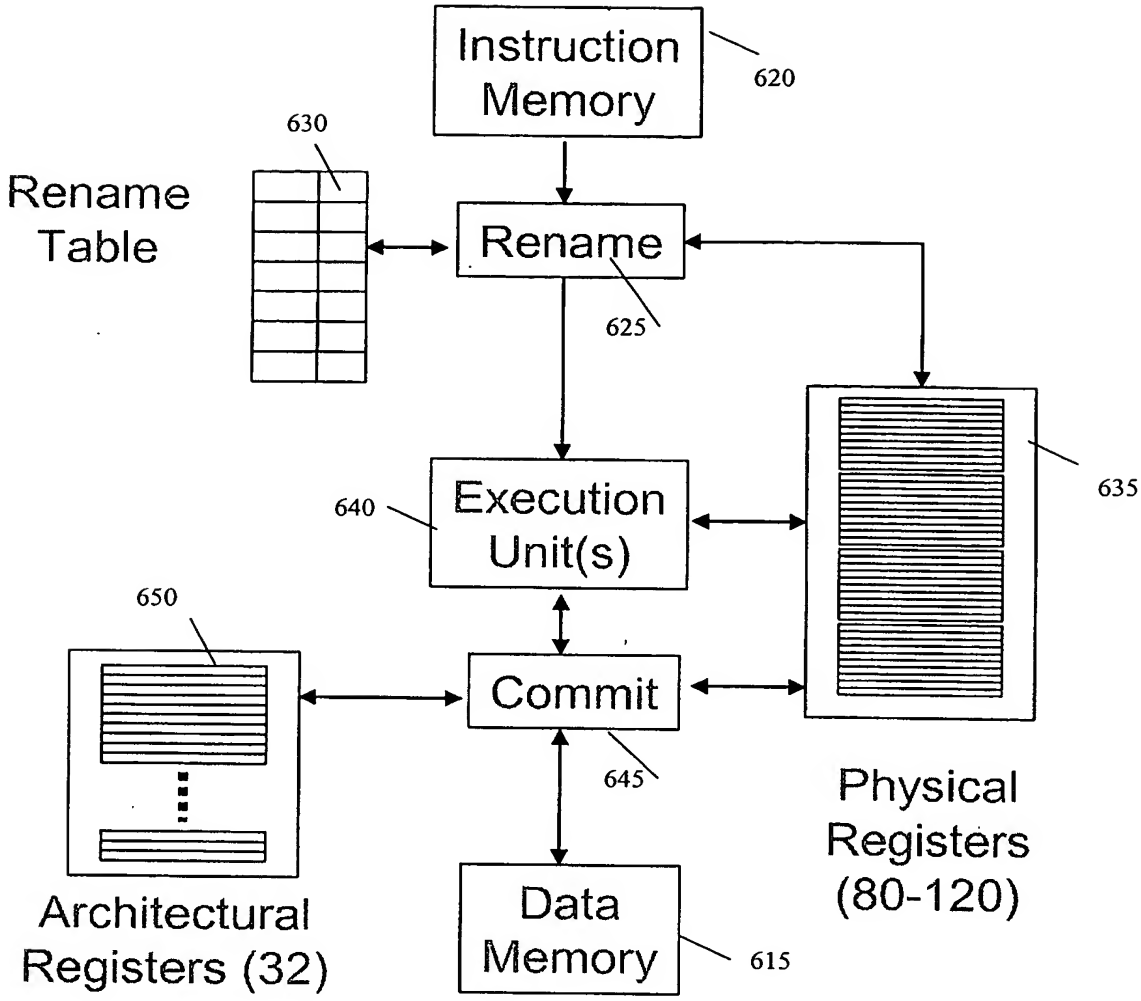
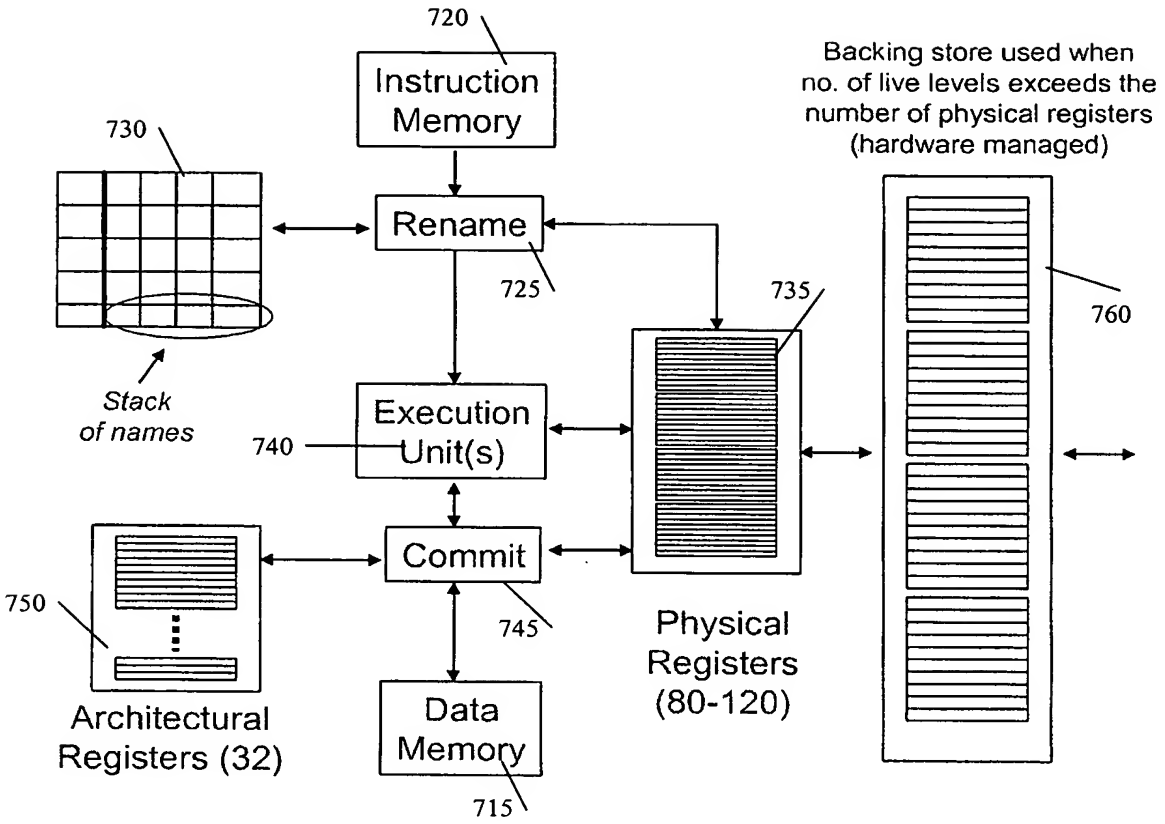

Instruction Memory
620
630
Rename Table
Rename
625
635
640
Execution Unit(s)
650
Commit
645
Physical Registers (80-120)
Architectural Registers (32)
Data Memory
615

Fig. 7


720
Instruction Memory
730
Rename
725
735
Backing store used when no. of live levels exceeds the number of physical registers (hardware managed)
760
Stack of names
740
Execution Unit(s)
Commit
745
750
Architectural Registers (32)
Data Memory
715
Physical Registers (80-120)